



Australian Government

Department of Defence
Science and Technology

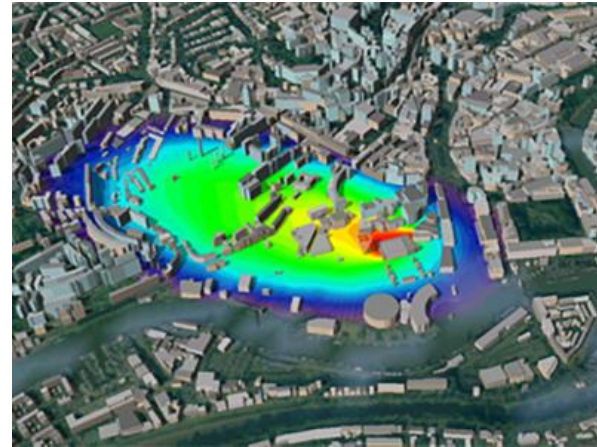
Estimation of Fields Using Binary Measurements From a Mobile Agent

Alex Leong, Mohammad Zamani

Defence Science and Technology Group, Australia

IEEE International Conference on Autonomous Systems, Aug. 2021

Introduction



- In the presence of a CBRN (chemical, biological, radiological, and nuclear) event, **mapping** out the **contaminated area / field** is an important task in allowing operations to be carried out
 - Humanitarian, disaster relief, military ...
 - Want to do it as **quickly** and **accurately** as possible

Outline

- Introduction
 - Mobile Autonomous Agents
 - System Model
- Field Estimation
- Active Sensing
- Simulation Results
- Conclusion



Previous work

- Source localization using mobile agents
 - Non-binary measurements [Ristic, Morelande, Gunatilaka, 2010]
 - Binary measurements, single source [Selvaratnam et.al., 2019]
- Field estimation with static sensors, binary measurements
 - [Battistelli et.al., 2019]
- Field estimation using mobile agents, non-binary measurements
 - [La, Sheng, Chen, 2015], [Razak, Sukumar, Chung, 2019]
- **Our work:** Field estimation using mobile agents, binary measurements
 - Concentrate on single agent case

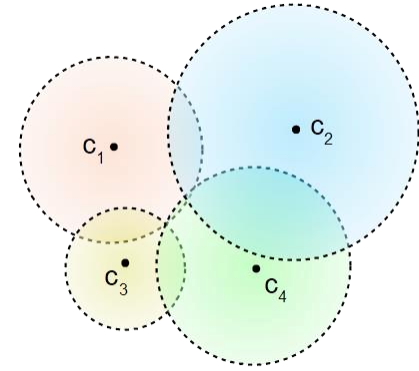
Measurement Model

- **Radiological** sources can be measured **pretty accurately**
- **Chemicals** used in attacks may be of **very low concentration**
- Current chemical sensing technologies **cannot give very precise readings** of such concentrations
 - Noisy and time varying
 - Sensor outputs could be one of several levels
- Assume **noisy and coarsely quantized measurements**
 - $z(\mathbf{x}) = q(\phi(\mathbf{x}) + v(\mathbf{x}))$, where \mathbf{x} is position, ϕ is field value, v is Gaussian noise, q is quantizer
 - In this work will consider special case of noisy **binary measurements** - reading is above or below a known **threshold** τ
 $z(\mathbf{x}) = \mathbb{1}(\phi(\mathbf{x}) + v(\mathbf{x}) > \tau)$ (algorithms will still hold for general case)

Field Model

- **Approximate** the field as a **sum of basis functions**

- Field model:
$$\phi(\mathbf{x}) = \sum_{j=1}^J \beta_j K_j(\mathbf{x})$$



where β_j are **weights**, $K_j(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{c}_j - \mathbf{x}\|^2}{\sigma_j^2}\right)$ are (radial) **basis functions**

- Used in works such as [Morelande, Skvortsov, 2009], [La, Sheng, Cheng, 2015], [Razak, Sukumar, Chung, 2019]
 - Mathematical results prove that for J large enough, can **approximate** many fields to **arbitrary accuracy**
- **Parameter estimation** approach: Pick a (large) J , choose \mathbf{c}_j 's and σ_j 's, and estimate the β_j 's

Field Estimation

- Field estimation problem reduces to problem of **estimating parameters** $\theta = (\beta_1, \dots, \beta_J, \sigma_v^2)$ where σ_v^2 is measurement noise variance
- We want to compute the **posterior pdf** $p(\theta | z_1, \dots, z_k; \mathbf{x}_1, \dots, \mathbf{x}_k)$ where z_k is the k -th measurement collected
 - Parameter estimates can then be derived from the posterior pdfs
- **Exact computation** of posterior pdfs is generally **intractable**
- Posterior pdfs can be **computed approximately** using **sequential Monte Carlo / particle filtering** techniques
 - Use approach of [Liu, West, 2001] suitable for estimation of constant parameters (rather than time-varying states)

Field Estimation

Algorithm 1 Sequential Monte Carlo algorithm for parameter estimation

- 1: **Algorithm Parameters:** $N \in \mathbb{N}$, $a \in (0, 1)$, $h = \sqrt{1 - a^2}$, $\eta \geq 0$, prior pdf $p_0(\boldsymbol{\theta})$
 - 2: **Inputs:** Measurement locations $\{\mathbf{x}_k\}$
 - 3: **Outputs:** Particles $\{\boldsymbol{\theta}_k^{(i)}\}$ and weights $\{\mathbf{w}_k^{(i)}\}$
 - 4: Sample particles $\boldsymbol{\theta}_0^{(i)}$, $i = 1, \dots, N$ from $p_0(\boldsymbol{\theta})$, and assign weights $\mathbf{w}_0^{(i)} = \frac{1}{N}$, $i = 1, \dots, N$
 - 5: **for** $k = 1, 2, \dots$, **do**
 - 6: Observe z_k at location \mathbf{x}_k
 - 7: **for** $i = 1, \dots, N$ **do**
 - 8: Compute $\mathbf{m}_{k-1}^{(i)} = a\boldsymbol{\theta}_{k-1}^{(i)} + (1-a)\bar{\boldsymbol{\theta}}_{k-1}$, where $\bar{\boldsymbol{\theta}}_{k-1} = \sum_{i=1}^N \mathbf{w}_{k-1}^{(i)} \boldsymbol{\theta}_{k-1}^{(i)}$
 - 9: Assign $\tilde{\mathbf{w}}_k^{(i)} \propto p(z_k | \mathbf{m}_{k-1}^{(i)}; \mathbf{x}_k) \mathbf{w}_{k-1}^{(i)}$
 - 10: **end for**
 - 11: Normalize $\{\tilde{\mathbf{w}}_k^{(i)}\}$ such that $\sum_{i=1}^N \tilde{\mathbf{w}}_k^{(i)} = 1$
 - 12: Sample N times with replacement a set of indices $\{i^- : i = 1, \dots, N\}$, from a distribution with probabilities $\mathbb{P}(i^- = j) = \tilde{\mathbf{w}}_k^{(j)}$
 - 13: **for** $i = 1, \dots, N$ **do**
 - 14: Sample a particle $\boldsymbol{\theta}_k^{(i)} \sim \mathcal{N}(\mathbf{m}_{k-1}^{(i^-)}, h^{2-\eta} \mathbf{V}_{k-1})$, where $\mathbf{V}_{k-1} = \sum_{i=1}^N \mathbf{w}_{k-1}^{(i)} (\boldsymbol{\theta}_{k-1}^{(i)} - \bar{\boldsymbol{\theta}}_{k-1})(\boldsymbol{\theta}_{k-1}^{(i)} - \bar{\boldsymbol{\theta}}_{k-1})^T$
 - 15: Assign weights $\mathbf{w}_k^{(i)} \propto \frac{p(z_k | \boldsymbol{\theta}_k^{(i)}; \mathbf{x}_k)}{p(z_k | \mathbf{m}_{k-1}^{(i^-)}; \mathbf{x}_k)}$
 - 16: **end for**
 - 17: Normalize $\{\mathbf{w}_k^{(i)}\}$ such that $\sum_{i=1}^N \mathbf{w}_k^{(i)} = 1$
 - 18: **end for**
-

Active Sensing

- **Given** a set of **measurements** (z_1, \dots, z_k) and their **locations** $(\mathbf{x}_1, \dots, \mathbf{x}_k)$ the particle filtering approach computes posterior pdfs and hence parameter estimates
- Can we “**optimize**” the **locations** in which measurements are made, to **reduce the time** needed to accurately **map the field**?
- **Active sensing** – actively choose locations for the sensor measurements, based on measurements currently collected
- One approach to active sensing is based on **Renyi divergence**
 - [Kreucher et.al, 2007], [Ristic, Morelande, Gunatilaka, 2010]

Active Sensing

- **Renyi divergence** between two pdfs $f_1(\cdot)$, $f_2(\cdot)$ defined as

$$D_\alpha(f_1||f_0) \triangleq \frac{1}{\alpha - 1} \ln \int f_1^\alpha(\mathbf{t}) f_0^{1-\alpha}(\mathbf{t}) d\mathbf{t}$$

- a measure of the **difference between two pdfs** (Kullback-Leibler divergence is a special case as $\alpha \rightarrow 1$)

- Approach to active sensing
 - look at **expected Renyi divergence** $\mathbb{E}[D_\alpha(p(\boldsymbol{\theta}|z_{1:k}; \mathbf{x}_{1:k})||p(\boldsymbol{\theta}|z_{1:k+1}; \mathbf{x}_{1:k+1}))]$ between posterior pdf at **current location** \mathbf{x}_k and posterior pdf at a set of **candidate future locations** \mathbf{x}_{k+1}
 - pick the **future location** which **maximizes** this
 - Intuition: Larger divergence means more “information” can potentially be obtained at the new location

Active Sensing

Algorithm 2 Active sensing algorithm: $\mathbf{x}_{k+1} = \text{ActiveSensing}(\mathbf{x}_k, \{\boldsymbol{\theta}_k^{(i)}\})$

- 1: **Algorithm Parameters:** $\varepsilon \geq 0$, $\alpha \in [0, \infty) \setminus \{1\}$, $\rho_0 \geq 0$, $N_\rho \in \mathbb{N}$, $N_d \in \mathbb{N}$, search region \mathcal{S}
- 2: **Inputs:** \mathbf{x}_k , $\{\boldsymbol{\theta}_k^{(i)}\}$
- 3: **Output:** Next measurement location \mathbf{x}_{k+1}
- 4: With probability ε set \mathbf{x}_{k+1} to a random location in \mathcal{S} , otherwise set

$$\mathbf{x}_{k+1} = \arg \max_{\mathbf{x}' \in \mathcal{X}_k} \frac{1}{\alpha - 1} \sum_{z_{k+1}=0}^1 \gamma_1(z_{k+1}|\mathbf{x}') \ln \frac{\gamma_\alpha(z_{k+1}|\mathbf{x}')}{(\gamma_1(z_{k+1}|\mathbf{x}'))^\alpha}$$

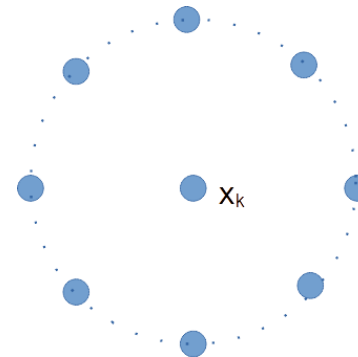
where

$$\mathcal{X}_k = \left\{ \mathbf{x}_k + \left(n\rho_0 \cos\left(\frac{2\pi\ell}{N_d}\right), n\rho_0 \sin\left(\frac{2\pi\ell}{N_d}\right) \right), \right. \\ \left. n = 0, \dots, N_\rho, \ell = 0, 1, \dots, N_d - 1 \right\} \cap \mathcal{S}$$

$$\gamma_\alpha(z_{k+1}|\mathbf{x}') = \frac{1}{N} \sum_{i=1}^N p(z_{k+1}|\boldsymbol{\theta}_k^{(i)}; \mathbf{x}')^\alpha$$

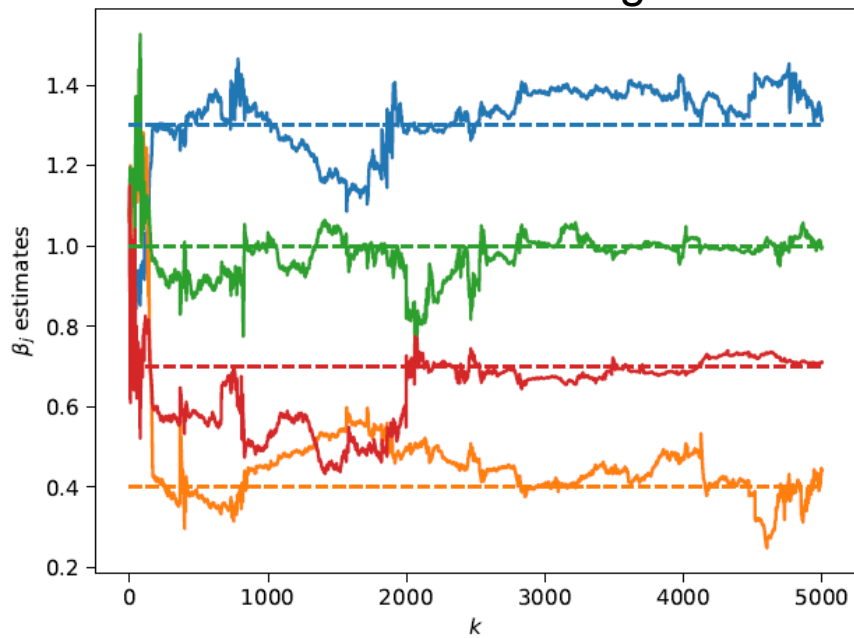
Simulation Studies – Example 1

- $J = 4$ basis functions
- True values of c_j 's and σ_j 's known
- **Candidate future locations** to optimize over in **active sensing** algorithm
 - Current location plus eight directions

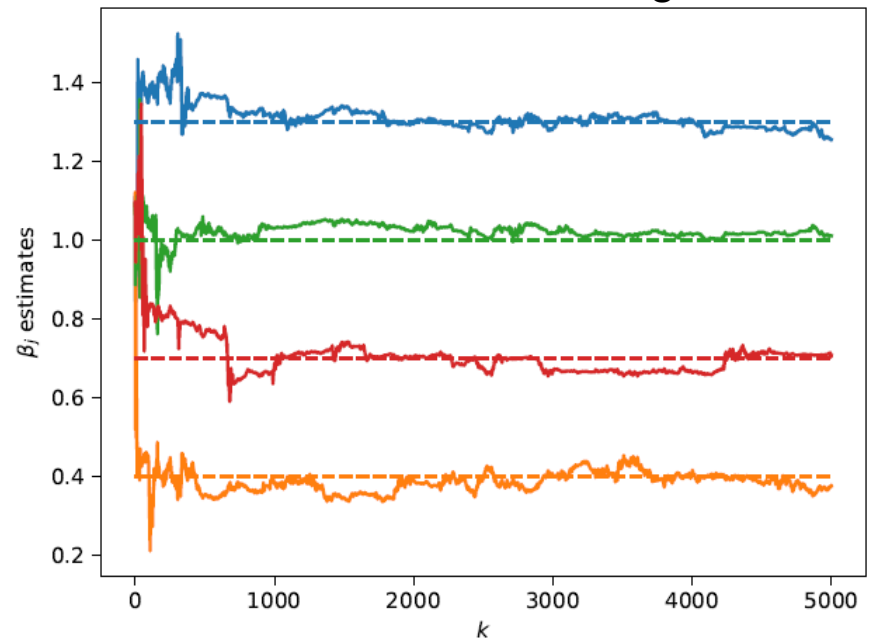


Simulation Studies – Example 1

No active sensing

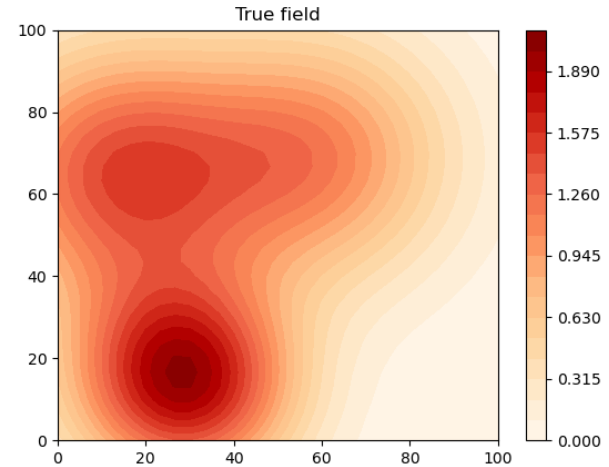


With active sensing

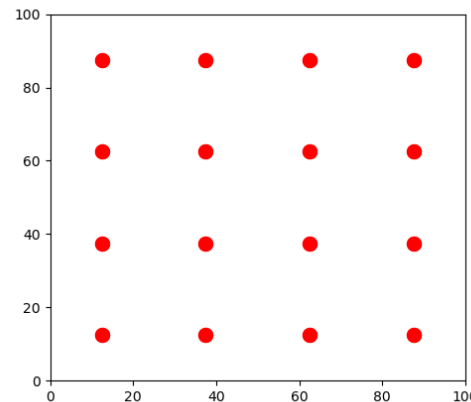


Simulation Studies – Example 2

- True field as shown
- True values of \mathbf{c}_j 's and σ_j 's not known



- For field estimation, use $J = 16$ basis functions, \mathbf{c}_j 's located on a "grid", $\sigma_j = 25, \forall j$



Simulation Results – Example 2

```
Ubuntu 18.04 (Snapshot 9) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminator
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://10.0.2.15:36149/
ros_comm version 1.14.10
SUMMARY
=====
PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.10
NODES
auto-starting new master
process[master]: started with pid [15799]
ROS_MASTER_URI=http://10.0.2.15:11311/
setting /run_id to 2bef55fc-b2d9-11eb-a3f1-080027294ad9
process[rosout-1]: started with pid [15810]
started core service [/rosout]
alex@alex-VirtualBox: ~/Distributed_swarm/Simulation_Catkin
roscore http://10.0.2.15:11311/204x23
alex@alex-VirtualBox: ~/Distributed_swarm/Simulation_Catkin 101x23
[ WARN ] [1620881114.818373962]: Received JointState is 1620877334.172326 seconds old.
[ WARN ] [1620881124.819137496]: Received JointState is 1620877335.133089 seconds old.
[ WARN ] [1620881134.828810522]: Received JointState is 1620877335.882778 seconds old.
[ WARN ] [1620881144.849965983]: Received JointState is 1620877336.463797 seconds old.
[ WARN ] [1620881154.851503695]: Received JointState is 1620877337.305425 seconds old.
[ WARN ] [1620881164.861388181]: Received JointState is 1620877337.835352 seconds old.
[ WARN ] [1620881174.872269689]: Received JointState is 1620877338.426234 seconds old.
[ WARN ] [1620881184.892170863]: Received JointState is 1620877339.086134 seconds old.
[ WARN ] [1620881194.907907365]: Received JointState is 1620877340.181871 seconds old.
[ WARN ] [1620881204.915865100]: Received JointState is 1620877340.729034 seconds old.
^C[husky_1/husky_1_laser_tf-9] killing on exit
[husky_1/husky_1_world_tf-8] killing on exit
[husky_1/twist_mux-7] killing on exit
[husky_1/robot_state_publisher-6] killing on exit
[husky_1/twist_marker_server-5] killing on exit
[husky_1/broadcast-4] killing on exit
[husky_1/ekf_localization-3] killing on exit
[husky_1/base_controller_spawner-2] killing on exit
[gazebo-1] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
alex@alex-VirtualBox: ~/Distributed_swarm/Simulation_Catkin$
alex@alex-VirtualBox: ~/Distributed_swarm/Swarm_Catkin 101x23
field measurements 0.7507241253044896 1.3697290107057798 1
[INFO] [1620881154.459074]: k=999, beta_hat=[ 0.71237087 0.20991344 1.11698535 0.27658499 1.25688589 0.32110188
1.16215942 0.09799654 -0.06392281 -0.36365867 0.54374373 0.36246546
-0.08087197 -0.21089496 0.39161212 -0.24028067], sigma_v_hat=0.380335
[INFO] [1620881154.602130]: field estimation done!
^C[husky_1/waypoint_follower-11] killing on exit
[husky_1/field_estimator-10] killing on exit
[husky_1/twist_controller-8] killing on exit
[husky_1/truthter-7] killing on exit
[husky_1/peer_sensor-5] killing on exit
[husky_1/plotter-6] killing on exit
[husky_1/peer_merge-4] killing on exit
[husky_1/get_field_meas_server-9] killing on exit[husky_1/filter-3] killing on exit
[plotter-2] killing on exit
true pose error: service [/gazebo/get_model_state] returned no response
peer sensor fake- my- error: service [/gazebo/get_model_state] returned no response
[comms-1] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
alex@alex-VirtualBox: ~/Distributed_swarm/Swarm_Catkin$
```


Conclusion

- **Field estimation** can be done even with **coarsely quantized / binary measurements**
- **Active sensing** mechanism can be incorporated into estimation algorithm

- **Extensions**
 - Multiple agents: Reduce the amount of time needed to estimate field, both centralized and decentralized schemes
 - Time-varying fields: Adapt approach of [Nemeth, Fearnhead, Mihaylova, 2014]